```c
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

typedef struct {
    int idSucursal;
    char nombreSucursal[25];
    char nombreProducto[25];
    char deporte[25];
    int stockProducto;
}stRegistro;

typedef struct
{
    int idSucursal;
    char nombreSucursal[25];
    char nombreProducto[25];
    char deporte[25];
    int stockProducto;
    struct nodo2 *anterior;
    struct nodo2 *siguiente;
}nodo2;


typedef struct
{
    nodo2 *primero;
    nodo2 *ultimo;
}Fila;


typedef struct{
    int idSucursal;
    char nombreSucursal[25];
}stSucursal;

typedef struct{
    char nombreProducto[25];
    char deporte[25];
    int stockProducto;
}stProducto;



typedef struct //Nodos lista simple
{
    stProducto producto;
    struct nodoProd *siguiente;
}nodoProd;

typedef struct //Celda
{
    stSucursal sucursal;
    nodoProd *productos;
}celdaSucursal;

stProducto crearSTProducto(char nombreProducto[25],char deporte[25],int stockProducto);
nodoProd *crearNodoProd(char nombreProducto[25],char deporte[25],int stockProducto);
nodoProd *agregarProd(nodoProd *lista,nodoProd *producto);
int buscarSucursal(celdaSucursal sucursales[],int idSucursal,int validos);
stSucursal nuevoSTSucursal(int idSucursal,char nombreSucursal[25]);
celdaSucursal crearSucursal(stSucursal sucursal);

void pasarACelda(celdaSucursal celdas[],Fila fila,int *validos);

Fila pasarAFila(char nombreArchivo[20]);
```

```
 67
 68  nodo2 *inicNodo2();
 69  nodo2 *crearNodo2(int idSucursal, char nombreSucursal[20],char nombreProducto[20],char deporte[25],int
stockProducto);
 70  nodo2 *agregarAlFinal(nodo2 *lista,nodo2 *nuevo);
 71  void mostrarNodo2(nodo2 *nodo);
 72  void mostrarNodo2Rec(nodo2 *nodo);
 73  void mostrarDato(nodo2 dato);
 74  void inicFila(Fila *fila);
 75
 76  void encolar(Fila *fila, nodo2 *nuevo);
 77  void mostrarFila(Fila fila);
 78
 79  void mostrarNodoprod(nodoProd *producto);
 80  void mostrarListaSimple(nodoProd *lista);
 81  void mostrarArregloSuc(celdaSucursal celdas[], int validos);
 82
 83  int buscarSucursalLD(celdaSucursal sucursales[], int validos, char nombreSucursal[30]);
 84  int sumatoriaRec(nodoProd *listaProductos,char deporte[30], char producto[30]);
 85
 86  void ejercicio1(Fila *fila,char nombreArchivo[20]);
 87  void ejercicio2(Fila fila);
 88  void ejercicio3(celdaSucursal celdas[],Fila fila,int *validos);
 89  void ejercicio4(celdaSucursal celdas[],int validos);
 90  void ejercicio5(celdaSucursal celdas[],int validos);
 91  //void pasarACSV(char archivoOrigen[40],char archivoDestino[40]);
 92
 93  int main()
 94  {
 95      Fila nuevaFila;
 96      inicFila(&nuevaFila);
 97      celdaSucursal celdas[200];
 98      int validos=0;
 99      char nombreArchivo[50]="archivoRegistrosIndumentaria.bin";
100      ejercicio1(&nuevaFila,nombreArchivo);
101      ejercicio2(nuevaFila);
102      ejercicio3(celdas,nuevaFila,&validos);
103      ejercicio4(celdas,validos);
104      ejercicio5(celdas,validos);
105
106      return 0;
107  }
108
109  nodo2 *inicNodo2()
110  {
111      return NULL;
112  }
113  nodo2 *crearNodo2(int idSucursal, char nombreSucursal[20],char nombreProducto[20],char deporte[25],int
stockProducto)
114  {
115      nodo2 *nuevo=(nodo2*)malloc(sizeof(nodo2));
116      nuevo->idSucursal=idSucursal;
117      strcpy(nuevo->nombreSucursal,nombreSucursal);
118      strcpy(nuevo->nombreProducto,nombreProducto);
119      strcpy(nuevo->deporte,deporte);
120      nuevo->stockProducto=stockProducto;
121      nuevo->anterior=NULL;
122      nuevo->siguiente=NULL;
123
124      return nuevo;
125  }
126
127  Fila pasarAFila(char nombreArchivo[20])
128  {
129      Fila cargado;
130      inicFila(&cargado);
```

```
131     FILE *archivo=fopen(nombreArchivo,"rb");
132     if(archivo)
133     {
134         nodo2 *nuevo;
135         stRegistro buffer;
136         while(fread(&buffer,sizeof(stRegistro),1,archivo)>0)
137         {
138             nuevo=crearNodo2(buffer.idSucursal,buffer.nombreSucursal,buffer.nombreProducto,buffer.deporte,
buffer.stockProducto);
139             encolar(&cargado,nuevo);
140         }
141         fclose(archivo);
142     }
143     else
144         printf("\n el archivo no existe/no se pudo abrir ");
145
146     return cargado;
147 }
148
149 nodo2 *agregarAlFinal(nodo2 *lista,nodo2 *nuevo)
150 {
151     if(!lista)
152     {
153         lista=nuevo;
154     }
155     else
156     {
157         lista->siguiente=nuevo;
158         nuevo->anterior=lista;
159     }
160     return nuevo;
161 }
162
163 void mostrarNodo2Rec(nodo2 *nodo)
164 {
165     if(nodo)
166     {
167         mostrarDato(*nodo);
168         mostrarNodo2Rec(nodo->siguiente);
169     }
170 }
171
172 void mostrarDato(nodo2 dato)
173 {
174         puts ("\n*********************************************************************************\n");
175         printf ("\nId de la Sucursal.....: %d \n", dato.idSucursal);
176         printf("\nNombre de la Sucursal: ……..….:%s \n", dato.nombreSucursal);
177         printf("\nNombre del Producto.....: %s \n ", dato.nombreProducto);
178         printf("\nDeporte al que pertenece.....: %s \n ", dato.deporte);
179         printf("\nStock del producto.....: %d \n", dato.stockProducto);
180         puts ("\n*********************************************************************************\n");
181 }
182
183
184 void inicFila(Fila *fila)
185 {
186     fila->primero=inicNodo2();
187     fila->ultimo=inicNodo2();
188 }
189
190 void encolar(Fila *fila, nodo2 *nuevo)
191 {
192     fila->ultimo=agregarAlFinal(fila->ultimo,nuevo);
193     if(!fila->primero)
194     {
195         fila->primero=fila->ultimo;
```

```
196     }
197 }
198
199
200 void ejercicio1(Fila *fila,char nombreArchivo[20])
201 {
202     *fila=pasarAFila(nombreArchivo);
203 }
204
205 void ejercicio2(Fila fila)
206 {
207     mostrarNodo2Rec(fila.primero);
208 }
209
210 /*void pasarACSV(char archivoOrigen[40],char archivoDestino[40])
211 {
212     FILE *aOrigen=fopen(archivoOrigen,"rb");
213     {
214         if(aOrigen)
215         {
216             FILE *aDestino=fopen(archivoDestino,"wb");
217
218             stRegistro buffer;
219
220             while(fread(&buffer,sizeof(stRegistro),1,aOrigen)>0)
221             {
222
fprintf(aDestino,"%i;%s;%s;%s;%i\n",buffer.idSucursal,buffer.nombreSucursal,buffer.nombreProducto,buffer.deporte
,buffer.stockProducto);
223             }
224             fclose(aDestino);
225             fclose(aOrigen);
226         }
227     }
228 }
229 */
230
231 stProducto crearSTProducto(char nombreProducto[25],char deporte[25],int stockProducto)
232 {
233     stProducto nuevoProducto;
234     strcpy(&nuevoProducto.nombreProducto,nombreProducto);
235     strcpy(&nuevoProducto.deporte,deporte);
236     nuevoProducto.stockProducto=stockProducto;
237     return nuevoProducto;
238 }
239
240 nodoProd *crearNodoProd(char nombreProducto[25],char deporte[25],int stockProducto)
241 {
242     nodoProd *nuevoNodoProd=(nodoProd*)malloc(sizeof(nodoProd));
243     nuevoNodoProd->siguiente=NULL;
244     nuevoNodoProd->producto=crearSTProducto(nombreProducto,deporte,stockProducto);
245
246     return nuevoNodoProd;
247 }
248
249 nodoProd *agregarProd(nodoProd *lista,nodoProd *producto)
250 {
251     if(!lista)
252     {
253         lista=producto;
254         //printf("\n %s %s %i
",producto->producto.deporte,producto->producto.nombreProducto,producto->producto.stockProducto);
255     }
256     else
257     {
258         nodoProd *iterador=lista;
```

```
259         nodoProd *anterior=iterador;
260         while(iterador->siguiente)
261         {
262             iterador=iterador->siguiente;
263         }
264         iterador->siguiente=producto;
265         //printf("\n %s %s %i
",producto->producto.deporte,producto->producto.nombreProducto,producto->producto.stockProducto);
266     }
267     return lista;
268 }
269
270 int buscarSucursal(celdaSucursal sucursales[],int idSucursal,int validos)
271 {
272     int ubicacion=-1; //-1 PARA DETERMINAR QUE NO SE ENCUENTRA EN NINGUNA PARTE
273     int contador=0;
274     while(ubicacion==-1 && contador<validos)
275     {
276         if(sucursales[contador].sucursal.idSucursal==idSucursal)
277         {
278             ubicacion=contador;
279         }
280         contador++;
281         //printf("\n contador %i");
282     }
283     return ubicacion;
284 }
285
286
287 stSucursal nuevoSTSucursal(int idSucursal,char nombreSucursal[25])
288 {
289     stSucursal nuevoSTSucursal;
290     nuevoSTSucursal.idSucursal=idSucursal;
291     strcpy(&nuevoSTSucursal.nombreSucursal,nombreSucursal);
292     return nuevoSTSucursal;
293 }
294
295 celdaSucursal crearSucursal(stSucursal sucursal)
296 {
297     celdaSucursal nuevaCelda;
298     nuevaCelda.sucursal.idSucursal=sucursal.idSucursal;
299     strcpy(&nuevaCelda.sucursal.nombreSucursal,sucursal.nombreSucursal);
300     nuevaCelda.productos=NULL;
301     return nuevaCelda;
302 }
303
304 void pasarACelda(celdaSucursal celdas[],Fila fila,int *validos)
305 {
306     int contador=0;
307     int ubicacion=-1;
308     nodo2 *listaSucursales=fila.primero;
309
310     while(listaSucursales)
311     {
312         ubicacion=buscarSucursal(celdas,listaSucursales->idSucursal,*validos);
313         while(ubicacion==-1 && contador<=*validos)
314         {
315             if(celdas[contador].sucursal.idSucursal==listaSucursales->idSucursal)
316             {
317                 ubicacion=contador;
318             }
319             contador++;
320         }
321         if(ubicacion==-1)//SI NO EXISTE LA SUCURSAL
322         {
323             stSucursal sucursal=nuevoSTSucursal(listaSucursales->idSucursal,listaSucursales->nombreSucursal
```

```
);
            celdaSucursal nuevaSucursal=crearSucursal(sucursal);
            celdas[*validos]=nuevaSucursal;

            stProducto nuevoProducto=crearSTProducto(listaSucursales->nombreProducto,listaSucursales->
deporte,listaSucursales->stockProducto);
            nodoProd *nuevoNodoProd=crearNodoProd(listaSucursales->nombreProducto,listaSucursales->deporte,
listaSucursales->stockProducto);
            celdas[*validos].productos=agregarProd(celdas[*validos].productos,nuevoNodoProd);
            *validos=*validos+1;
        }
        else //SI ENCUENTRA LA SUCURSAL
        {
            stProducto nuevoProducto=crearSTProducto(listaSucursales->nombreProducto,listaSucursales->
deporte,listaSucursales->stockProducto);
            nodoProd *nuevoNodoProd=crearNodoProd(listaSucursales->nombreProducto,listaSucursales->deporte,
listaSucursales->stockProducto);
            celdas[ubicacion].productos=agregarProd(celdas[ubicacion].productos,nuevoNodoProd);
        }
        ubicacion=-1;
        listaSucursales=listaSucursales->siguiente;

    }
}

void ejercicio3(celdaSucursal celdas[],Fila fila,int *validos)
{
    pasarACelda(celdas,fila,validos);
}

void mostrarNodoprod(nodoProd *producto)
{
    if(producto)
    {
        printf("\n %-10s | %-20s | %i ",producto->producto.deporte, producto->producto.nombreProducto,
producto->producto.stockProducto);
    }
}

void mostrarListaSimple(nodoProd *lista)
{
    if(lista)
    {
        nodoProd *iterador=lista;
        while(iterador)
        {
            mostrarNodoprod(iterador);
            iterador=iterador->siguiente;
        }
    }
    else
    {
        printf("\nLa lista esta vacia! ");
    }
}

void mostrarArregloSuc(celdaSucursal celdas[], int validos)
{
    if(validos>=0)
    {
        printf("\n Celdas Validas %i ",validos);
        for(int x=0;x<validos;x++)
        {
            printf("\n\n");
            printf("\n SUCURSAL ID: %-2i | NOMBRE: %-10s",celdas[x].sucursal.idSucursal,celdas[x].sucursal.
nombreSucursal);
```

```
383             printf("\n\n");
384             mostrarListaSimple(celdas[x].productos);
385         }
386     }
387 }
388 
389 void ejercicio4(celdaSucursal celdas[],int validos)
390 {
391     mostrarArregloSuc(celdas,validos);
392 }
393 
394 
395 int buscarSucursalLD(celdaSucursal sucursales[], int validos, char nombreSucursal[30])
396 {
397     int ubicacion=-1;
398     int contador=0;
399     while(contador < validos && ubicacion==-1)
400     {
401         if(strcmpi(sucursales[contador].sucursal.nombreSucursal,nombreSucursal)==0)
402         {
403             ubicacion=contador;
404         }
405         contador++;
406     }
407     return ubicacion;
408 }
409 
410 /*
411 
412 int buscarSucursalLD(celdaSucursal sucursales[], int validos, char nombreSucursal[30])
413 {
414     int ubicacion=-1;
415     int contador=0;
416     while(contador < validos && ubicacion==-1)
417     {
418         if(strstr(sucursales[contador].sucursal.nombreSucursal,nombreSucursal)!=NULL)
419         {
420             ubicacion=contador;
421         }
422         contador++;
423     }
424     return ubicacion;
425 }
426 
427 */
428 
429 int sumatoriaRec(nodoProd *listaProductos,char deporte[30], char producto[30])
430 {
431     int resultado=0;
432     if(listaProductos)
433     {
434         if(strstr(listaProductos->producto.deporte,deporte)!=NULL && (strstr(listaProductos->producto.
nombreProducto,producto))!=NULL)
435         {
436             resultado=listaProductos->producto.stockProducto + sumatoriaRec(listaProductos->siguiente,
deporte,producto);
437         }
438     }
439     return resultado;
440 }
441 
442 void ejercicio5(celdaSucursal celdas[],int validos)
443 {
444     char producto[30];
445     char deporte[30];
446     char sucursal[30];
```

```
447     int sumatoria=0;
448     int posSucursal=-1;
449
450     printf("\nIngrese el nombre de la sucursal a buscar: ");
451     gets(sucursal);
452
453     posSucursal=buscarSucursalLD(celdas,validos,sucursal);
454
455     if(posSucursal==-1)
456     {
457         printf("\n La sucursal no existe!");
458     }
459     else
460     {
461         printf("\n Ingrese el deporte: ");
462         fflush(stdin);
463         //gets(deporte);
464         scanf("%s",deporte);
465
466         printf("\n Ingrese el articulo: ");
467         fflush(stdin);
468         scanf("%s",producto);
469         //gets(producto);
470
471         sumatoria=sumatoriaRec(celdas[posSucursal].productos,deporte,producto);
472     }
473
474     printf("\nLa sumatoria es de : %i",sumatoria);
475 }
```